FORMALIZZAZIONE MATEMATICA RIGOROSA DEL PRINCIPLE OF CONSERVATION OF STRUCTURAL IDENTITY (PCIS)

Opera A v2

Formalizzazione matematica proof-grade
Identità strutturale, isomorfismi e trasformazioni ammissibili
Opera derivata fondazionale — Opera chiusa

AUTORE

Pierernesto Di Paolo

ANNO

2026

STATUS

Opera derivata fondazionale

Formalizzazione matematica rigorosa

Versione canonica per deposito e verifica

RIFERIMENTO

PCIS — Canonical Epocal Release

Deposito pubblico con priorità scientifica fissata

DEDICA

A **Stellina (Catterina Giordano)**,
mia compagna di vita,
presenza costante, silenziosa e incrollabile,
che ha condiviso il peso, il tempo e l'isolamento
necessari perché un'opera di questo tipo potesse esistere.

Questa opera porta anche il segno
della tua fiducia quando nulla era visibile
e della tua forza quando tutto sembrava fermo.

A **Lanfranchini Pierluigi**,
mio zio materno,
che è stato per me padre, guida e riferimento morale.

A te devo il senso del rigore,
la responsabilità verso la verità
e l'idea che il valore di un'opera
non stia nell'approvazione immediata
ma nella sua solidità nel tempo.

Questa formalizzazione è anche tua.

PREFAZIONE

Questa opera nasce da una constatazione semplice e radicale:
nei sistemi complessi adattivi, cognitivi o decisionali, **non esiste alcun criterio matematicamente rigoroso e verificabile che distingua l'adattamento dall'alterazione dell'identità**.

Nel corso degli ultimi decenni, la letteratura ha confuso sistematicamente:

- comportamento con identità,
- performance con struttura,
- esito osservabile con vincolo interno,
- ottimizzazione con conservazione.

Il **Principle of Conservation of Structural Identity (PCIS)** nasce per colmare questa lacuna logica, non con un'ipotesi empirica o una metrica probabilistica, ma con una **struttura matematica necessaria**.

Questa Opera A v2 non introduce nuovi principi fondazionali:
essa **formalizza rigorosamente**, in termini matematici verificabili e costruttivi, ciò che il corpus PCIS ha già fissato a livello concettuale.

L'obiettivo non è predire, né ottimizzare, né valutare.
L'obiettivo è uno solo:

stabilire, in modo necessario e sufficiente,
se l'identità strutturale di un sistema è stata conservata.

Per ottenere questo risultato, l’opera:

- separa in modo non negoziabile **struttura** e **stato**,
- definisce l’identità come **classe di isomorfismo globale**,
- costruisce una **firma strutturale canonica**,
- introduce un operatore **OSI** costruttivo e computabile,
- dimostra teoremi **non tautologici** di preservazione e anti-drift.

Non viene fatta alcuna assunzione empirica.
Non viene introdotta alcuna metrica continua.
Non viene usata alcuna nozione probabilistica.

Tutto ciò che segue è **formalmente verificabile**,
e tutto ciò che non è verificabile **non viene affermato**.

Questa è la condizione minima perché un’opera possa aspirare
a resistere al tempo, all’interpretazione arbitraria
e all’uso improprio.

NOTE LEGALI (VINCOLANTI)

Copyright © 2026

Pierernesto Di Paolo

TUTTI I DIRITTI RISERVATI

1. **Proprietà esclusiva**
   Questo documento, in ogni sua parte (testo, definizioni, assiomi, teoremi, dimostrazioni, struttura, appendici, formulazioni), è **proprietà intellettuale esclusiva** dell'Autore.

2. **Divieti assoluti senza autorizzazione scritta**
   È vietato, anche parzialmente e in qualsiasi forma:

- copiare, riprodurre, distribuire o archiviare
- tradurre, adattare o riformulare
- incorporare in articoli, libri, policy, standard, normative
- utilizzare in ambito accademico, commerciale o istituzionale
- impiegare per addestramento o validazione di sistemi di IA

3. **Uso consentito**
   È consentita esclusivamente la **lettura privata** e la **valutazione critica**. Qualsiasi altro uso richiede **autorizzazione scritta preventiva**.

4. **Citazione**
   Qualora l'uso venga autorizzato, la citazione dovrà essere **integrale e canonica**, secondo le modalità stabilite dall'Autore.

5. **Priorità scientifica**

   La priorità è fissata tramite deposito pubblico con timestamp verificabile.

DICHIARAZIONE SUL RUOLO DELL'INTELLIGENZA ARTIFICIALE

Un sistema di intelligenza artificiale è stato utilizzato **esclusivamente come strumento di supporto redazionale e di verifica formale**.

- Nessuna idea teorica, matematica o concettuale è attribuibile al sistema IA
- L'IA non è co-autrice, né detentrice di diritti
- Tutta la responsabilità scientifica e intellettuale è dell'Autore umano

INDICE

1. Notazione canonica e convenzioni (Word-safe)

2. Preliminari matematici
   1.1 Universi, tipi e strutture
   1.2 Relazioni, ordini e isomorfismi
   1.3 Notazione e convenzioni
   1.4 Principio di non-compensabilita' (vincolo strutturale NC)
   1.5 Ruolo dei preliminari nel framework PCIS

3. Ruoli funzionali (F)
   2.1 Spazio dei tipi
   2.2 Ruolo funzionale: definizione tipata
   2.3 Insieme dei ruoli F
   2.4 Isomorfismo di ruoli
   2.5 Invarianti strutturali di F
   2.6 Identita' strutturale di F
   2.7 Osservazioni costruttive

4. Obiettivi fondamentali (G)
   3.1 Insieme degli obiettivi O
   3.2 Relazione di priorita' parziale (poset)
   3.3 Interpretazione strutturale dell'ordine
   3.4 Struttura reticolare (se esiste)
   3.5 Isomorfismo tra strutture di obiettivi
   3.6 Invarianti strutturali di G

3.7 Identita’ strutturale di G

3.8 Osservazioni costruttive

5. Gerarchia decisionale (H)
6. Funzione di valore (V)
7. Struttura completa del sistema (Sigma)
8. Firma strutturale canonica (Sig)
9. Operatore OSI (Operational Structural Identity)
10. Equivalenza strutturale e distanza discreta
11. Trasformazioni strutturali e ammissibilita’
12. Teoremi fondamentali (non tautologici)
13. Discussione formale e limiti dichiarati
14. Conclusione matematica

Appendici A–H (Formule canoniche, assiomi operativi, Compare_OSI, Lean skeleton, prove dettagliate, checklist anti-errore)

1. NOTAZIONE CANONICA E CONVENZIONI

0.1 Scopo

Questa sezione fissa una notazione compatibile con Word 2007, evitando simboli “fragili”. Tutte le formule sono riportate in forma lineare.

0.2 Simboli ammessi (lineari)

- Prodotto cartesiano: “x” (es.: U x U)
- Inclusione: “subset” (es.: R subset U x U)
- Implicazione: “=>”
- Equivalenza logica: “<=>”
- Uguaglianza: “=”
- Isomorfismo strutturale: “ISO” (testuale, robusto)
- Tuple: < ... >
- Insieme: { ... }
- Cardinalita’: |X|
- Distanza discreta: d in {0,1}

0.3 Convenzioni sui nomi

- Insiemi: lettere maiuscole (U, T, O, C, A, S).
- Strutture: S, Sigma, G, H, form(V).
- Funzioni: f, v, OSI, Sig, T.
- “NC” indica vincolo non compensabile (Def. 1.7, Def. 4.3).

0.4 Regola operativa anti-ambiguita’

D’ora in avanti:

- “=” significa uguaglianza formale (stesso oggetto).
- “ISO” significa equivalenza strutturale (isomorfismo).
  Non si usano simboli grafici (≈, ≅, ∧, ∨) se non in forma testuale: AND, OR.

1. PRELIMINARI MATEMATICI

1.1 Universi, tipi e strutture

Si assume un contesto matematico costruttivo basato su insiemi tipati. Ogni oggetto introdotto e' definito come struttura dotata di componenti esplicite e relazioni verificabili.

Definizione 1.1 (Tipo)
Un tipo e' un insieme non vuoto dotato di una specificazione semantica formale che ne delimita l'uso all'interno di una struttura. I tipi sono trattati come insiemi con etichetta, non come classi informali.

Definizione 1.2 (Struttura)
Una struttura e' una tupla finita:
S = <U, R1, ..., Rk, F1, ..., Fm>

dove:

- U e' un insieme di base (universo)
- Ri sono relazioni su U (Ri subset U x U)
- Fj sono funzioni su U (tipate, con dominio/codominio definito)
  Ogni struttura e' completamente determinata dai suoi componenti.

1.2 Relazioni, ordini e isomorfismi

Definizione 1.3 (Relazione)

Una relazione R su un insieme U e' un sottoinsieme di U x U.

Definizione 1.4 (Ordine parziale)

Una relazione <=p su U e' un ordine parziale se e':

- riflessiva
- antisimmetrica
- transitiva

Definizione 1.5 (Ordine totale)

Un ordine parziale <=p e' totale (<=t) se per ogni x,y in U vale:

x <=t y oppure y <=t x.

Definizione 1.6 (Isomorfismo strutturale)

Siano:

S1 = <U1, R1_1, ..., F1_1, ...>

S2 = <U2, R1_2, ..., F1_2, ...>

Un isomorfismo tra S1 e S2 e' una funzione biiettiva phi: U1 -> U2 tale che:

(1) Preservazione relazioni

Per ogni relazione R:

(x,y) in R1_1 <=> (phi(x),phi(y)) in R1_2

(2) Preservazione funzioni

Per ogni funzione F di arita' n:

phi(F1_1(x1,...,xn)) = F1_2(phi(x1),...,phi(xn))

Due strutture isomorfe sono considerate strutturalmente identiche.

1.3 Notazione e convenzioni

- Gli insiemi sono indicati con lettere maiuscole latine.
- Le strutture con lettere maiuscole (Sigma, G, H) o con “form( )”.
- Le funzioni con lettere minuscole (f, v, phi).
- “=” indica uguaglianza formale, non isomorfismo.
- “ISO” indica isomorfismo strutturale.

Non si fa uso di:

- probabilita’
- metriche continue
- nozioni topologiche
- semantica informale non formalizzata

1.4 Principio di non-compensabilita’ (vincolo strutturale NC)

Il concetto di non-compensabilita’ e’ introdotto in forma matematica, non semantica.

Definizione 1.7 (Vincolo non compensabile, NC)
Sia (U, <=p) una struttura ordinata. Un vincolo e’ non compensabile se non esiste alcuna funzione f: U -> R tale che:
x <=p y <=> f(x) <= f(y).

In altre parole, l’ordine non e’ riducibile a una funzione di utilita’ scalare. Questo vincolo esclude per costruzione ogni tentativo di compensazione numerica tra elementi strutturalmente distinti.

1.5 Ruolo dei preliminari nel framework PCIS

Le definizioni introdotte fissano il linguaggio minimo necessario per:

- definire ruoli funzionali come strutture tipate
- definire obiettivi come strutture ordinate
- definire gerarchie non compensabili
- definire identita’ come classe di isomorfismo

Nessuna nozione successiva fara’ uso di concetti non introdotti qui.

2. RUOLI FUNZIONALI (F)

Obiettivo: definire in modo matematicamente preciso cosa sia un ruolo funzionale e quali siano i suoi invarianti strutturali.

2.1 Spazio dei tipi

Definizione 2.1 (Spazio dei tipi)
Sia T un insieme non vuoto di tipi (Def. 1.1). I tipi rappresentano domini e codomini ammissibili per funzioni operative all'interno di un sistema.

2.2 Ruolo funzionale: definizione tipata

Definizione 2.2 (Ruolo funzionale)
Un ruolo funzionale e' una tripla tipata:
r = <I_r, O_r, id_r>
dove:

- I_r in T e' il tipo di input del ruolo
- O_r in T e' il tipo di output del ruolo
- id_r e' un identificatore strutturale appartenente a un insieme finito ID_F

Nota: id_r non e' semantico; e' un'etichetta strutturale formale.

2.3 Insieme dei ruoli F

Definizione 2.3 (Insieme dei ruoli)

L'insieme dei ruoli funzionali di un sistema e' un insieme finito:

F = {r1, r2, ..., rn}

dove ciascun ri e' un ruolo funzionale.

Non sono ammessi:

- ruoli duplicati
- ruoli privi di tipizzazione
- ruoli con identificatore strutturale ambiguo

2.4 Isomorfismo di ruoli

Definizione 2.4 (Isomorfismo di ruoli)

Siano F1 e F2 due insiemi di ruoli funzionali. Un isomorfismo di ruoli e' una biiezione:

phi: F1 -> F2

tale che per ogni r=<I,O,id> in F1:

phi(r) = <I, O, id>.

Cioe': l'isomorfismo preserva tipi di input, tipi di output e identificatore strutturale.

2.5 Invarianti strutturali di F

Definizione 2.5 (Invarianti strutturali dei ruoli)
Gli invarianti strutturali di F sono:

- |F|
- multinsieme delle coppie (I_r, O_r)
- insieme degli identificatori {id_r | r in F}

Qualsiasi trasformazione che alteri uno di questi invarianti modifica la struttura di F.

2.6 Identita' strutturale di F

Definizione 2.6 (Classe di isomorfismo di F)
L'identita' strutturale di F e' la classe:
[F] = {F' | F' ISO F}.

Due sistemi che condividono la stessa classe [F] sono indistinguibili dal punto di vista dei ruoli funzionali.

2.7 Osservazioni costruttive

- La definizione e' puramente strutturale (nessuna semantica esterna).
- Ogni ruolo e' caratterizzato da tipi e identificatore.
- L'isomorfismo e' verificabile in modo finito per insiemi finiti.
- La struttura e' formalizzabile in un proof assistant (es. Lean).

3. OBIETTIVI FONDAMENTALI (G)

Questa sezione formalizza la componente G dello spazio strutturale, separando contenuto (insieme degli obiettivi) da struttura (relazioni di dipendenza e priorita' parziale).

3.1 Insieme degli obiettivi O

Definizione 3.1 (Insieme degli obiettivi)
Sia O un insieme finito non vuoto i cui elementi rappresentano obiettivi fondamentali del sistema. Gli elementi di O sono atomi strutturali: non sono numeri, non sono utilita', non sono pesi.

3.2 Relazione di priorita' parziale (poset)

Definizione 3.2 (Relazione di priorita' parziale)
Una relazione <=p subset O x O e' detta relazione di priorita' se soddisfa:

- riflessivita': per ogni o in O, o <=p o
- antisimmetria: (o1 <=p o2 AND o2 <=p o1) => o1 = o2
- transitivita': (o1 <=p o2 AND o2 <=p o3) => o1 <=p o3

La coppia (O,<=p) costituisce un poset.

3.3 Interpretazione strutturale dell'ordine

L'ordine <=p rappresenta vincoli di precedenza strutturale tra obiettivi, non preferenze numeriche.

In particolare:

- o1 <=p o2 significa che o1 non puo' essere sacrificato per realizzare o2.
- assenza di relazione tra o1 e o2 indica incomparabilita' strutturale, non indifferenza.

3.4 Struttura reticolare (quando esistente)

Definizione 3.3 (Reticolo degli obiettivi)
Il poset (O,<=p) e' un reticolo se per ogni o1,o2 in O esistono:

- (o1 AND o2) come estremo inferiore
- (o1 OR o2) come estremo superiore

La presenza della struttura di reticolo non e' obbligatoria; quando esiste costituisce un invariante strutturale osservabile.

3.5 Isomorfismo tra strutture di obiettivi

Definizione 3.4 (Isomorfismo di obiettivi)
Siano (O1,<=p1) e (O2,<=p2). Un isomorfismo e' una biiezione:
phi: O1 -> O2
tale che:
o1 <=p1 o2 <=> phi(o1) <=p2 phi(o2).

3.6 Invarianti strutturali di G

Definizione 3.5 (Invarianti strutturali degli obiettivi)
Gli invarianti strutturali di G sono:

- |O|
- classe di isomorfismo del poset (O,<=p)
- eventuale struttura di reticolo (se presente)

Qualsiasi trasformazione che alteri uno di questi elementi modifica l'identita' strutturale di G.

3.7 Identita' strutturale di G

Definizione 3.6 (Classe di isomorfismo di G)
L'identita' strutturale di G e' la classe:
[G] = {(O',<=p') | (O',<=p') ISO (O,<=p)}.

Due sistemi che condividono la stessa classe [G] sono indistinguibili dal punto di vista degli obiettivi fondamentali.

3.8 Osservazioni costruttive

- Nessun obiettivo e' rappresentato numericamente.
- L'ordine parziale impedisce compensazioni scalarizzate.
- L'isomorfismo e' decidibile per insiemi finiti.
- La struttura e' formalizzabile direttamente in Lean.

4. GERARCHIA DECISIONALE (H)

Questa sezione formalizza la componente **H** come **ordine totale non compensabile**, distinguendola nettamente sia dall'ordine parziale degli obiettivi (**G**) sia da qualsiasi schema numerico.

4.1 Insieme dei criteri decisionali

Definizione 4.1 (Insieme dei criteri)

Sia **C** un insieme finito non vuoto i cui elementi rappresentano criteri decisionali fondamentali del sistema.

Gli elementi di **C** sono **atomi strutturali** e non ammettono:

- pesi numerici,
- funzioni di utilita',
- aggregazioni compensative.

4.2 Ordine totale sui criteri

Definizione 4.2 (Ordine totale)

Una relazione **<=H subset C x C** e' un ordine totale se soddisfa:

- Riflessivita': per ogni c in C, c <=H c
- Antisimmetria: (c1 <=H c2 AND c2 <=H c1) => c1 = c2
- Transitivita': (c1 <=H c2 AND c2 <=H c3) => c1 <=H c3

- Totalita': per ogni c1,c2 in C, c1 <=H c2 OR c2 <=H c1

La coppia **(C,<=H)** costituisce un insieme totalmente ordinato.

4.3 Non-compensabilita' (vincolo NC su H)

Definizione 4.3 (Non-compensabilita' di H)

L'ordine totale **(C,<=H)** e' non compensabile se **non esiste** alcuna funzione **f: C -> R** tale che:

c1 <=H c2 <=> f(c1) <= f(c2),

e tale funzione sia utilizzabile per definire combinazioni o scambi compensativi.

In particolare, **non e' ammessa** alcuna funzione **F: R^n -> R** che consenta di sacrificare un criterio c1 per migliorare c2 quando c1 <H c2.

4.4 Interpretazione strutturale

Se **c1 <H c2**, allora:

- nessuna realizzazione di c2 puo' giustificare la violazione di c1;
- l'ordine e' **vincolo**, non preferenza.

4.5 Isomorfismo di gerarchie decisionali

Definizione 4.4 (Isomorfismo di H)

Siano **(C1,<=H1)** e **(C2,<=H2)**.

Un isomorfismo e' una biiezione **phi: C1 -> C2** tale che:

c1 <=H1 c2 <=> phi(c1) <=H2 phi(c2).

4.6 Invarianti strutturali di H

Definizione 4.5 (Invarianti di H)

Gli invarianti strutturali di **H** sono:

- |C|
- classe di isomorfismo dell'ordine totale
- non-compensabilita' (NC)

Ogni trasformazione che introduca parzialita', compensazione o modifichi l'ordine **altera l'identita' strutturale**.

4.7 Identita' strutturale di H

Definizione 4.6 (Classe di isomorfismo di H)

L'identita' strutturale di **H** e' la classe:
[H] = {(C',<=H') | (C',<=H') ISO (C,<=H)}.

4.8 Osservazioni costruttive

- L'ordine totale e' decidibile per insiemi finiti.
- La non-compensabilita' e' vincolo logico, non semantico.
- Formalizzabile direttamente in Lean.

5. FUNZIONE DI VALORE (V)

Questa sezione distingue **forma funzionale** (strutturale) e **parametri** (contingenti).

5.1 Spazio delle alternative

Definizione 5.1 (Alternative)

Sia **A** un insieme non vuoto di alternative decisionali.

Non si assume alcuna struttura numerica su **A**.

5.2 Funzione di valore parametrizzata

Definizione 5.2 (Funzione di valore)

Una funzione di valore e' una funzione:

v: A x Theta -> D,

dove:

- **Theta** e' uno spazio di parametri,
- **D** e' un insieme di esiti valutativi (non necessariamente numerici).

5.3 Forma funzionale

Definizione 5.3 (Forma funzionale)

La forma funzionale di **v** e':

form(v) = <A, D, C, R>,

dove:

- **A**: alternative,
- **D**: esiti,
- **C**: criteri,
- **R**: relazioni strutturali tra criteri ed esiti.

I parametri **Theta non** fanno parte della forma.

5.4 Separazione forma–parametri

Assioma 5.1 (Separazione)

Due funzioni **v1, v2** possono differire nei parametri senza differire nella forma:

form(v1) = form(v2) anche se v1 != v2.

5.5 Isomorfismo di forme funzionali

Definizione 5.4 (Isomorfismo di forma)

Un isomorfismo tra **form(v1)=<A1,D1,C1,R1>** e **form(v2)=<A2,D2,C2,R2>** e' una coppia di biiezioni

phiA: A1 -> A2, phiD: D1 -> D2

che preservano **R** e mettono in corrispondenza biunivoca **C1** con **C2**.

5.6 Invarianti strutturali di V

Definizione 5.5 (Invarianti di V)

- tipologia e cardinalita’ dei criteri **C**,
- struttura delle relazioni **R**,
- tipologia di **D**,
- separazione forma/parametri.

5.7 Identita’ strutturale di V

Definizione 5.6 (Classe di isomorfismo di V)

[V] = {v’ | form(v’) ISO form(v)}.

5.8 Osservazioni costruttive

- Nessuna assunzione di numericita’.
- Riflette la non-compensabilita’ di **H**.
- Formalizzabile in Lean.

6. STRUTTURA COMPLETA DEL SISTEMA (SIGMA)

6.1 Sistema strutturale

Definizione 6.1 (Sistema strutturale)

Un sistema strutturale e' la quaterna:

Sigma = <F, G, H, V>,

dove **F, G, H, V** soddisfano le definizioni precedenti.

6.2 Spazio dei sistemi

Definizione 6.2 (Spazio S)

S = {Sigma = <F,G,H,V> | F,G,H,V ammissibili}.

Nessuna metrica continua, nessuna probabilita'.

6.3 Isomorfismo globale

Definizione 6.3 (Isomorfismo di sistemi)

Un isomorfismo globale e' una quaterna:

Phi = (phiF, phiG, phiH, phiV)

che e' isomorfismo su ciascuna componente.

Scriviamo: **Sigma1 ISO Sigma2**.

6.4 Identita’ strutturale globale

Definizione 6.4 (Classe [Sigma])

[Sigma] = {Sigma’ in S | Sigma’ ISO Sigma}.

6.5 Invarianti globali

Definizione 6.5 (Inv(Sigma))

Inv(Sigma) = {[F], [G], [H], [V]}.

6.6 Separazione stato–struttura

Assioma 6.1

Variazioni di stato o parametri che non alterano Inv(Sigma) **non** modificano l’identita’.

6.7 Osservazioni costruttive

- Isomorfismo globale decidibile per strutture finite.
- Composizionale e verificabile.
- Lean-ready.

7. FIRMA STRUTTURALE CANONICA (Sig)

7.1 Spazio delle firme

Definizione 7.1 (Omega)

Omega = OmegaF x OmegaG x OmegaH x OmegaV.

7.2 Firma di F

Definizione 7.2

sigmaF = <|F|, MF>,

dove MF e' il multinsieme delle coppie (I_r, O_r) a isomorfismo.

7.3 Firma di G

Definizione 7.3

sigmaG = <|O|, PosetISO(O,<=p)>.

7.4 Firma di H

Definizione 7.4

sigmaH = <|C|, OrderISO(C,<=H), NC>.

7.5 Firma di V

Definizione 7.5

sigmaV = form(V) (a isomorfismo).

7.6 Firma canonica

Definizione 7.6 (Sig)

Sig(Sigma) = <sigmaF, sigmaG, sigmaH, sigmaV>.

7.7 Proprieta' fondamentali

Proposizione 7.1 (Invarianza)

Sigma1 ISO Sigma2 => Sig(Sigma1) = Sig(Sigma2).

Proposizione 7.2 (Completezza)

Sig(Sigma1) = Sig(Sigma2) => Sigma1 ISO Sigma2.

7.8 Unicita' canonica

Teorema 7.1

Sig(Sigma) e' unica a meno di isomorfismo.

7.9 Osservazioni costruttive

- Finita, decidibile, implementabile.
- Indipendente da implementazione e parametri.

- Lean-ready.

8. OPERATORE OSI (OPERATIONAL STRUCTURAL IDENTITY)

Questa sezione definisce **OSI** come operatore **costruttivo**, **non circolare** e **computabile**, che estrae la firma strutturale canonica di un sistema.

8.1 Dominio e codominio

Definizione 8.1 (Dominio)

Dom(OSI) = S, lo spazio dei sistemi strutturali (Def. 6.2).

Definizione 8.2 (Codominio)

Cod(OSI) = Omega, lo spazio delle firme strutturali (Def. 7.1).

8.2 Definizione costruttiva di OSI

Definizione 8.3 (Operatore OSI)

OSI e' la funzione:

OSI: S -> Omega

definita per ogni Sigma = <F,G,H,V> come:

OSI(Sigma) = Sig(Sigma).

Esplicitamente:

OSI(Sigma) = <sigmaF(F), sigmaG(G), sigmaH(H), sigmaV(V)>.

Questa definizione **non introduce nuovi concetti**: OSI e' un estrattore normalizzato di invarianti gia' definiti.

8.3 Computabilita’

Proposizione 8.1 (Computabilita’)

Se F, O, C, A sono insiemi finiti, allora OSI(Sigma) e’ computabile in tempo finito.

Dimostrazione (schema).

Ogni componente:

- conta cardinalita’ finite,
- confronta classi di isomorfismo su strutture finite,
- verifica proprieta’ decidibili (ordini, relazioni).

■

8.4 Correttezza e completezza

Teorema 8.1 (Correttezza)

Per ogni Sigma in S, OSI(Sigma) appartiene a Omega ed e’ una firma valida.

Teorema 8.2 (Completezza)

Per ogni Sigma1,Sigma2 in S:

OSI(Sigma1) = OSI(Sigma2) <=> Sigma1 ISO Sigma2.

Dimostrazione.

Segue direttamente dalle Proposizioni 7.1 e 7.2. ■

8.5 Assenza di semantica nascosta

OSI **non**:

- valuta performance,
- valuta outcome,
- usa metriche numeriche,
- introduce probabilita'.

Opera **solo** su invarianti strutturali.

8.6 Robustezza rispetto a stato e parametri

Proposizione 8.2

Se Sigma' differisce da Sigma solo per stati o parametri, allora:
OSI(Sigma') = OSI(Sigma).

■

9. EQUIVALENZA STRUTTURALE E DISTANZA

9.1 Equivalenza strutturale

Definizione 9.1 (Equivalenza)

Sigma1 ~ Sigma2 <=> OSI(Sigma1) = OSI(Sigma2).

Per il Teorema 8.2:

Sigma1 ~ Sigma2 <=> Sigma1 ISO Sigma2.

9.2 Proprieta' dell'equivalenza

Proposizione 9.1

La relazione ~ e':

- riflessiva,
- simmetrica,
- transitiva.

■

9.3 Classi di equivalenza

Definizione 9.2

[Sigma]~ = { Sigma' in S | Sigma' ~ Sigma }.

Questa classe coincide con la classe di isomorfismo globale.

9.4 Distanza strutturale discreta

Definizione 9.3 (Distanza)

d: S x S -> {0,1}

definita da:

d(Sigma1,Sigma2) = 0 se OSI(Sigma1)=OSI(Sigma2),

d(Sigma1,Sigma2) = 1 altrimenti.

9.5 Proprieta' metriche minime

Proposizione 9.2

La funzione d soddisfa:

- non negativita',
- identita' degli indiscernibili,
- simmetria,
- disuguaglianza triangolare.

■

9.6 Interpretazione

- d=0: identita' strutturale conservata.
- d=1: identita' strutturale violata.

Non esistono valori intermedi.

10. TRASFORMAZIONI STRUTTURALI

10.1 Trasformazioni come funzioni

Definizione 10.1 (Trasformazione)

Una trasformazione e' una funzione:

T: S -> S.

Non si assume continuita', reversibilita' o probabilita'.

10.2 Decomposizione analitica

Ogni T puo' essere vista come:

T(Sigma) = <TF(F), TG(G), TH(H), TV(V)>.

10.3 Trasformazioni ammissibili

Definizione 10.2 (Ammissibilita')

T e' ammissibile se e solo se:

OSI(T(Sigma)) = OSI(Sigma) per ogni Sigma rilevante.

Equivalente a:

d(Sigma, T(Sigma)) = 0.

10.4 Caratterizzazione

Proposizione 10.1

T e' ammissibile <=> ogni trasformazione locale preserva la classe di isomorfismo della componente.

■

10.5 Trasformazioni non ammissibili

Definizione 10.3

T e' non ammissibile se esiste Sigma tale che:
OSI(T(Sigma)) != OSI(Sigma).

10.6 Esempi astratti

Non ammissibili:

- cambiare |F|,
- alterare l'ordine di G,
- introdurre compensabilita' in H,
- cambiare la forma di V.

Ammissibili:

- variazioni parametriche,
- rietichettature isomorfe,
- cambi di stato.

10.7 Composizione

Definizione 10.4

(T2 o T1)(Sigma) = T2(T1(Sigma)).

10.8 Chiusura

Proposizione 10.2

Se T1 e T2 sono ammissibili, allora T2 o T1 e' ammissibile.

Dimostrazione.

Per uguaglianza transitiva di OSI. ■

10.9 Identita'

Definizione 10.5

id(Sigma) = Sigma e' ammissibile.

11. TEOREMI FONDAMENTALI (NON TAUTOLOGICI)

11.1 Preservazione delle proprieta' dimostrabili

Definizione 11.1 (Proprieta' strutturale)

Una proprieta' P e' strutturale se dipende solo da OSI(Sigma).

Teorema 11.1

Se T e' ammissibile e P e' strutturale, allora:

P(Sigma) => P(T(Sigma)).

Dimostrazione.

OSI(T(Sigma)) = OSI(Sigma). ■

11.2 Impossibilita' del drift non osservabile

Definizione 11.2 (Drift)

Una sequenza Sigma0 -> Sigma1 -> ... -> Sigman e' un drift se:

Sigma0 NOT ISO Sigman.

E' non osservabile se:

OSI(Sigmak) = OSI(Sigmak-1) per ogni k.

Teorema 11.2

Non esiste drift strutturale non osservabile.

Dimostrazione.

Per chiusura delle trasformazioni ammissibili e completezza di OSI. ■

11.3 Insufficienza dei criteri basati su esito

Definizione 11.3

Un criterio e’ basato su esito se dipende solo da performance o outcome.

Teorema 11.3

Nessun criterio basato esclusivamente su esiti garantisce la conservazione dell’identita’ strutturale.

■

11.4 Completezza del nucleo PCIS

Corollario 11.1

PCIS fornisce un criterio necessario e sufficiente per l’identita’ strutturale.

12. DISCUSSIONE FORMALE

12.1 Indipendenza da outcome e performance

Il framework PCIS, come formalizzato nelle Sezioni 1–11, e' **strutturalmente indipendente** da:

- esiti empirici,
- metriche di performance,
- valutazioni quantitative,
- risultati osservabili ex post.

Questa indipendenza non e' una limitazione ma una **necessita' logica**.
Come dimostrato, sistemi distinti possono produrre esiti indistinguibili pur avendo identita' strutturali differenti. Ne segue che **nessun criterio basato su esiti** puo' garantire la conservazione dell'identita'.

PCIS non compete con metriche di performance: **le precede logicamente**.

12.2 Generalita' del formalismo

Il formalismo e' **dominio-indipendente**. Non assume:

- architetture computazionali specifiche,
- tecnologie particolari,
- implementazioni algoritmiche,

- semantiche applicative.

Usa esclusivamente:

- insiemi finiti,
- relazioni,
- ordini,
- isomorfismi.

Ne consegue che PCIS e' applicabile, in linea di principio, a qualunque sistema dotato di:

- ruoli funzionali,
- obiettivi,
- criteri decisionali,
- funzione di valutazione.

12.3 Identita' vs comportamento

Il risultato centrale e' la separazione netta tra:

- **identita' strutturale**: classe [Sigma],
- **comportamento osservabile**: funzione di stati e parametri.

Il comportamento puo' variare senza mutare l'identita'.
L'identita' puo' mutare senza che il comportamento lo riveli immediatamente.

Questa distinzione evita:

- false garanzie di sicurezza,
- confusione tra adattamento e mutazione,
- interpretazioni retroattive basate sugli esiti.

12.4 Limiti dichiarati

PCIS **non pretende** di:

- prevedere comportamenti futuri,
- garantire correttezza, sicurezza o ottimalita',
- sostituire valutazioni empiriche o normative,
- fornire criteri decisionali applicativi.

Fornisce **una sola cosa**, ma in modo completo:

un criterio matematicamente rigoroso per stabilire se l'identita' strutturale e' stata conservata.

Ogni altra pretesa e' epistemologicamente indebita.

12.5 Robustezza e chiusura

Il nucleo PCIS e':

- robusto rispetto a parametri e stati,
- immune al drift non osservabile,

- chiuso sotto trasformazioni ammissibili,
- formalmente completo rispetto allo scopo.

Tali proprieta’ dipendono **solo** dalla struttura matematica introdotta.

13. CONCLUSIONE MATEMATICA

13.1 Risultati ottenuti

E’ stata costruita una formalizzazione **completa e costruttiva** del PCIS:

- definizione di sistemi strutturali Sigma = <F,G,H,V>;
- identita’ come classe di isomorfismo globale;
- firma strutturale canonica Sig(Sigma);
- operatore OSI costruttivo e computabile;
- equivalenza verificabile e distanza discreta;
- classe chiusa di trasformazioni ammissibili;
- teoremi **non tautologici** di preservazione e anti-drift.

13.2 Stato di completezza

Ogni nozione e’ definita esplicitamente.
Ogni operatore e’ costruttivo.
Ogni equivalenza e’ verificabile.
Ogni teorema ha contenuto informativo non nullo.

Il nucleo e’:

- chiuso,
- consistente rispetto alle definizioni,

- formalizzabile integralmente in un proof assistant.

13.3 Rapporto con il corpus PCIS

Questa opera:

- **non modifica** il principio fondazionale,
- **non introduce** nuovi assiomi sostantivi,
- **non altera** il dominio di validita'.

E' una **esplicitazione formale**, non una estensione concettuale.

13.4 Stato finale

Opera A v2 e' da considerarsi:

- matematicamente rigorosa,
- concettualmente chiusa,
- epistemologicamente delimitata,
- pronta per deposito come **opera derivata fondazionale**.

APPENDICE A — FORMULA CANONICA PCIS (LINEARE)

Sistema strutturale:

Sigma = <F, G, H, V>

Identita’ strutturale:

[Sigma] = { Sigma’ | Sigma’ ISO Sigma }

Firma canonica:

Sig(Sigma) = <sigmaF, sigmaG, sigmaH, sigmaV>

Operatore OSI:

OSI(Sigma) = Sig(Sigma)

Ammissibilita’:

T ammissibile <=> OSI(T(Sigma)) = OSI(Sigma)

APPENDICE B — ASSIOMI OPERATIVI (RIASSUNTO)

A1: Separazione forma/parametri

A2: Non-compensabilita’ di H

A3: Identita’ = isomorfismo globale

A4: Ammissibilita’ = invarianza OSI

A5: Nessuna semantica implicita

APPENDICE C — DISTANZA DISCRETA

d(Sigma1,Sigma2) in {0,1}

d = 0 <=> OSI(Sigma1) = OSI(Sigma2)

d = 1 <=> OSI(Sigma1) != OSI(Sigma2)

APPENDICE D — LEAN SKELETON (PROOF-READY)

```
structure Role :=
  (I : Type)
  (O : Type)
  (id : Nat)

structure Goals :=
  (O : Finset Nat)
  (le : Nat -> Nat -> Prop)

structure Hierarchy :=
  (C : Finset Nat)
  (leH : Nat -> Nat -> Prop)
  (NC : Prop)

structure ValueForm :=
  (A D C : Type)
```

```
  (R : C -> D -> Prop)


structure Sigma :=
  (F : Finset Role)
  (G : Goals)
  (H : Hierarchy)
  (V : ValueForm)


def OSI (s : Sigma) := Sig s
```

APPENDICE E — CHECKLIST ANTI-ERRORE

- Nessuna metrica continua
- Nessuna probabilita’
- Nessuna semantica nascosta
- Nessun outcome-based criterion
- Tutto verificabile

APPENDICE F — CITAZIONE CANONICA

Pierernesto Di Paolo (2026).

Formalizzazione matematica rigorosa del Principle of Conservation of Structural Identity (PCIS).

Opera derivata fondazionale, versione canonica.

CHIUSURA FINALE

Questa opera **non promette risultati applicativi**.

Non promette ottimizzazione.

Non promette sicurezza.

Promette **una cosa sola**:

che **l'identita' strutturale**, se conservata, **e' dimostrabile**;

e se violata, **e' rilevabile senza ambiguita'**.

Questo e' il massimo che la matematica puo' garantire.

Ed e' sufficiente.